BioPig: Developing Cloud Computing Applications for Next-Generation Sequence Analysis

Karan Bhatia, Zhong Wang
Joint Genomics Institute
Lawrence Berkeley National Laboratory

Next Generation sequencing is producing ever larger data sizes with a growth rate outpacing Moore's Law. The data deluge has made many of the current sequence analysis tools obsolete because they do not scale with data. Here we present BioPig, a collection of cloud computing tools to scale data analysis and management. Pig is a flexible data scripting language that uses Apache's Hadoop data structure and map reduce framework to process very large data files in parallel and combine the results. BioPig extends Pig with capability with sequence analysis. We will show the performance of BioPig on a variety of bioinformatics tasks, including screening sequence contaminants, Illumina QA/QC, and gene discovery from metagenome data sets using the Rumen metagenome as an example.

Application

kmer histograms

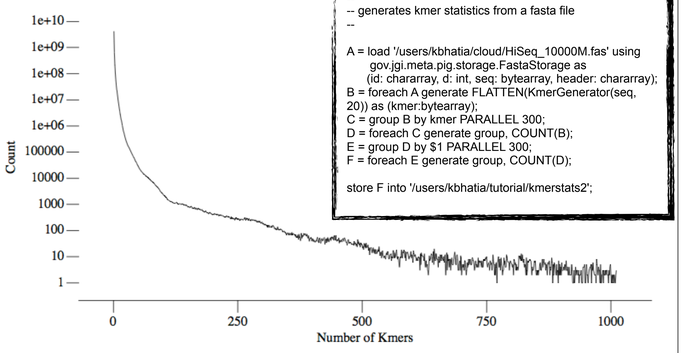
Kmers (also sometimes called ngrams) represent a sliding window of fixed length (*k*) across the sequence. Kmers are quite versatile and are the basis for a variety of analytics. A Kmer Histogram is a plot of the frequencies of kmers across a dataset that is an indicator for coverage, the existance of metagenomes or sequencing errors. The below plot shows typical *meta*genomic kmer frequencies for one lane of Illumina HiSeq with k=20.

Computationally, Kmer histograms require significant memory to maintain a hash table of counts. However, kmer generation and counting is easily parallelized using Hadoop MapReduce.

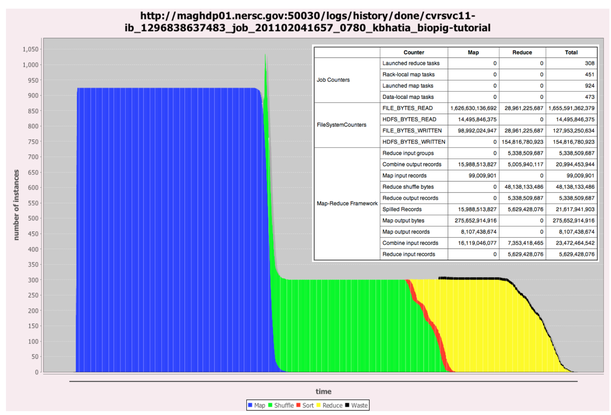
```
--
-- generates kmer statistics from a fasta file
--

A = load '/users/kbhatia/cloud/HiSeq_10000M.fas' using
      gov.jgi.meta.pig.storage.FastaStorage as
    (id: chararray, d: int, seq: bytearray, header: chararray);
B = foreach A generate FLATTEN(KmerGenerator(seq,
20)) as (kmer:bytearray);
C = group B by kmer PARALLEL 300;
D = foreach C generate group, COUNT(B);
E = group D by $1 PARALLEL 300;
F = foreach E generate group, COUNT(D);

store F into '/users/kbhatia/tutorial/kmerstats2';
```



Running 1000-way parallelism generates 300GB of intermediate data, 5 Billion kmers, and completes in 15 min.



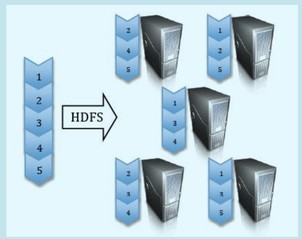
	Counter	Map	Reduce	Total
Job Counters	Launched reduce tasks	0	0	308
	Rack-local map tasks	0	0	451
	Launched map tasks	0	0	924
	Data-local map tasks	0	0	473
FileSystemCounters	FILE_BYTES_READ	1,626,630,136,692	28,961,225,687	1,655,591,362,379
	HDFS_BYTES_READ	14,495,846,375	0	14,495,846,375
	FILE_BYTES_WRITTEN	98,992,024,947	28,961,225,687	127,953,250,634
	HDFS_BYTES_WRITTEN	0	154,816,780,923	154,816,780,923
Map-Reduce Framework	Reduce input groups	0	5,338,509,687	5,338,509,687
	Combine output records	15,988,513,827	5,005,940,117	20,994,453,944
	Map input records	99,009,901	0	99,009,901
	Reduce shuffle bytes	0	48,138,133,486	48,138,133,486
	Reduce output records	0	5,338,509,687	5,338,509,687
	Spilled Records	15,988,513,827	5,829,428,076	21,817,941,903
	Map output bytes	275,652,914,916	0	275,652,914,916
	Map output records	8,107,438,674	0	8,107,438,674
	Combine input records	16,119,046,077	7,353,418,465	23,472,464,542
	Reduce input records	0	5,829,428,076	5,829,428,076

Performance scales linearly with size of sequences, and linearly with cluster size. Double the data, double the nodes.

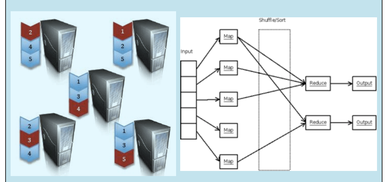
Amazon Elastic MapReduce provides similar performance with new CC.Large nodes (@$2/hr each)

Cloud Technology

HDFS



MapReduce



Pig

Pig Democratizes Large-scale Data Analysis

- The Pig version is:
  - **5% of the code**
  - **5% of the time**
  - Within 50% of the execution time.
- Innovation increasingly driven from large-scale data analysis
  - Need fast iteration to understand the *right questions*
  - More minds contributing = more value from your data

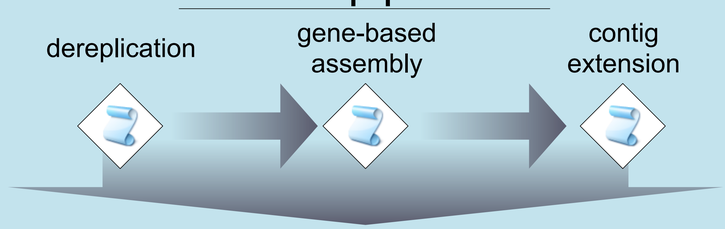
BioPig

features

- Custom data loaders for FASTA and FASTQ data files
- Wrapped many external Bioinformatics applications: Blast, Velvet, Newbler, CAP3
- native support for KMER generation
- supports paired and unpaired sequences
- Additional functions for N50, subsequence sequence neighbors

- leverage all the existing capabilities of Pig including data joins, filters, aggregate, etc.

rumen pipeline



http://codaset.com/zhongwang/meta

Application

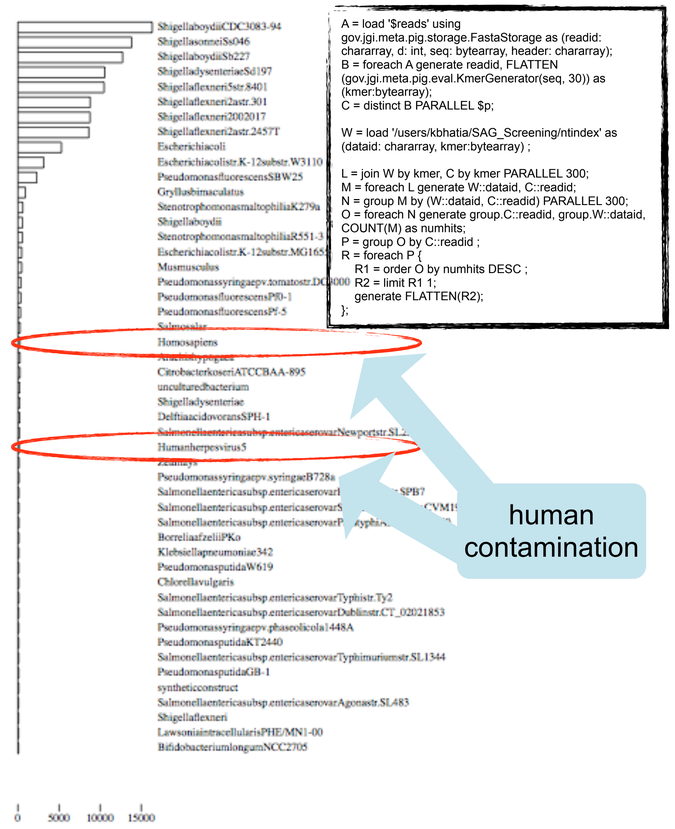
Screening

Metagenomic sequence data can represent a multitude of individual species found at the sample site, as well as species introduced in the sequencing process. Especially valuable for Single-Cell sequencing, kmer-based pairwise matching to the NT database can identifiy contaminants, as shown here.

```
A = load '$reads' using
gov.jgi.meta.pig.storage.FastaStorage as (readid:
chararray, d: int, seq: bytearray, header: chararray);
B = foreach A generate readid, FLATTEN
(gov.jgi.meta.pig.eval.KmerGenerator(seq, 30)) as
(kmer:bytearray);
C = distinct B PARALLEL $p;

W = load '/users/kbhatia/SAG_Screening/ntindex' as
(dataid: chararray, kmer:bytearray) ;

L = join W by kmer, C by kmer PARALLEL 300;
M = foreach L generate W::dataid, C::readid;
N = group M by (W::dataid, C::readid) PARALLEL 300;
O = foreach N generate group.C::readid, group.W::dataid,
COUNT(M) as numhits;
P = group O by C::readid ;
R = foreach P {
    R1 = order O by numhits DESC ;
    R2 = limit R1 1;
    generate FLATTEN(R2);
};
```



Magellan Cloud Testbed

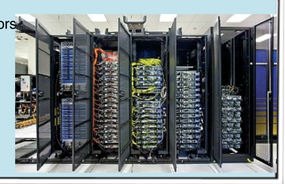
Base Compute Nodes
- 560 nodes
- 2 quad-core Intel Nehalem 2.67 GHz processors per node
- 8 cores per node (4,480 total cores)
- 24 GB DDR3 1333 MHz memory per node

Expanded Compute Nodes
- 160 nodes - all being used for Magellan research
- 2 quad-core Intel Nehalem 2.67 GHz processors per node
- 8 cores per node (1,280 total cores)
- 48 GB DDR3 1066 MHz memory per node
- 1 TB (local) SATA disk per node

Login/Network Service Nodes
- 18 nodes
- 2 quad-core Intel Nehalem 2.67 GHz processors
- 8 cores per node (144 total cores)
- 48 GB DDR3 1066 MHz memory per node

High Performance Interconnect
- 4X QDR InfiniBand, fibre optic cables
- Local fat-trees with a global 2D mesh

Cooling
- Liquid Cooled



The work conducted by the U.S. Department of Energy Joint Genome Institute is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

DISCLAIMER